Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №4**

Дисциплина: «Разработка профессиональных приложений»

Классы

Вариант №11

Выполнил:

студент группы ИВТАСбд-22

Курдюк В.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

Задание по варианту

Необходимо переделать лабораторную работу №3 с использованием классов, описывающих предметную область, заданную вариантом, с реализацией следующих особенностей:

1. Класс должен содержать итератор

2. Должна быть реализована перегрузка стандартных операций (repr, например)

3. Должно быть реализовано наследование

4. Запись значений в свойства - только через \_\_setattr\_\_

5. Возможность доступа к элементам коллекции по индексу (\_\_getitem\_\_)

6. Должны быть реализованы статические методы

7. Должны быть реализованы генераторы

**Описание реализации**

Данный код представляет собой реализацию таблицы, которая содержит классы Posts и Table.

Класс Posts является наследником класса Table. Он содержит методы работы со строками таблицы, такие как их добавление, удаление, сортировка, поиск и т.д. Также класс Table имеет стандартную реализацию перегрузки операторов (repr, str), что упрощает работу с объектами этого класса. Доступ к свойствам объекта контролируется через метод setattr и метод getattr.

Класс Table описывает саму таблицу, которая содержит объекты класса Posts. Класс Table имеет методы для работы с таблицей в целом: добавление строк, удаление строк, сортировка и т.д. Кроме того, класс Table реализует методы итерации через объекты класса Posts (с помощью iter и next) и доступ по индексу(getitem).

Код также содержит примеры использования всех реализованных функций - например, сортировка и фильтрация данных из файла input.csv, изменение значения поля cost в строке таблицы, сохранение таблицы в файл и т.д.

**Описание возникших затруднений**

Организация связей между классами: для того, чтобы строка таблицы могла быть членом таблицы и использовать ее методы, необходимо организовать связь между этими классами.

Разработка тестов: для проверки корректности работы классов необходимо было разработать соответствующие тесты, которые покрывают все возможные сценарии использования классов.

Описание альтернативных способов решения

Использование функционального программирования вместо объектно-ориентированного подхода, чтобы уменьшить количество кода и повысить читаемость.

Добавление обработки ошибок при выполнении операций с таблицей, таких как удаление несуществующей строки или неверный формат файла с данными.

Разделение методов класса Table на более мелкие и специализированные методы для упрощения поддержки и тестирования кода. Применение контекстных менеджеров для безопасного открытия и закрытия файлов с данными.

Тестирование

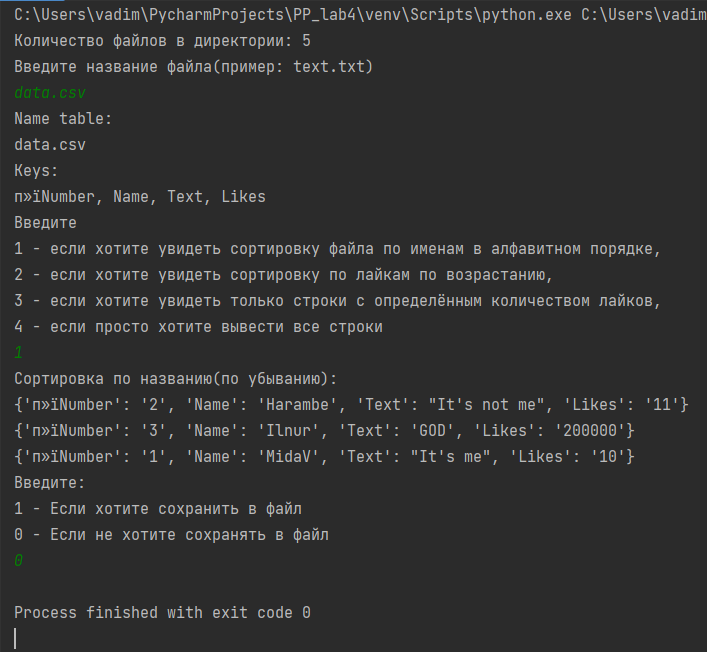


Рис1. Входные данные

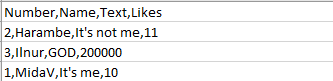


Рис2. Результат работы программы

Вывод

В данной лабораторной работе были приобретены базовые навыки объектно-ориентированного программирования на языке Python, в частности в работе со таблицами, вводом и выводом файлов и преобразованием типов данных.

Код программы

import os

import csv

class Table:

def \_\_init\_\_(self):

self.mylist = {}

self.name\_file = None

self.i = 0

self.n = 0

def set\_attribute(self, name\_file):

self.\_\_setattr\_\_("name\_file", name\_file)

def read(self):

with open(name\_file, newline='') as file:

reader = csv.DictReader(file, delimiter=";")

self.mylist = [row for row in reader]

@staticmethod

def directory():

print("Количество файлов в директории:", len(os.listdir(path=".")))

def \_\_iter\_\_(self):

return self

def \_\_next\_\_(self):

self.n = len(self.mylist)

if self.i < self.n:

i = self.mylist[self.i]

self.i += 1

return i

else:

raise StopIteration

def \_\_repr\_\_(self):

return "Name table: \n" + name\_file + "\nKeys:\n" + ", ".join(i for i in self.mylist[0])

class Posts(Table):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.new\_list = None

Table.read(self)

def sort\_cost(self):

sign = input("Введите знак с сравнения. Вот список знаков: >, <, >=, <=, ==, !=\n")

quantity = int(input("Введите количество с которым хотите сравнивать\n"))

print("Вывод по условию(стоимость", sign, " ", quantity, ": ")

for x in self.mylist:

if sign == '>':

if int(x["Likes"]) > quantity:

print(x)

elif sign == '<':

if int(x["Likes"]) < quantity:

print(x)

elif sign == '!=':

if x["Likes"] != quantity:

print(x)

elif sign == '>=':

if int(x["Likes"]) >= quantity:

print(x)

elif sign == '<=':

if int(x["Likes"]) <= quantity:

print(x)

elif sign == '==':

if x["Likes"] == quantity:

print(x)

def sort\_name(self):

print("Сортировка по названию(по убыванию):")

self.mylist = sorted(self.mylist, key=lambda x: x['Name'])

Posts.printlist(self.mylist)

def sort\_like(self):

print("Сортировка по стоимости(по убыванию):")

self.mylist = sorted(self.mylist, key=lambda x: x['Likes'])

self.mylist = list(reversed(self.mylist))

Posts.printlist(self.mylist)

def printlist(self):

for x in self:

print(x)

def save\_file(self, check):

if check == 1:

fields = ["п»їNumber", "Name", "Text", "Likes"]

with open('output.csv', 'w', newline='') as file:

writer = csv.DictWriter(file, fieldnames=fields)

writer.writeheader()

for row in self.mylist:

writer.writerow(row)

print("Сохранение завершено")

Table.directory()

name\_file = input("Введите название файла(пример: text.txt)\n")

post = Posts()

post.set\_attribute(name\_file)

print(post)

key = int(

input("Введите \n1 - если хотите увидеть сортировку файла по именам в алфавитном порядке, \n2 - если хотите "

"увидеть сортировку по лайкам по возрастанию, \n3 - если хотите увидеть только строки с определённым "

"количеством лайков, \n4 - если просто хотите вывести все строки\n")

)

if key == 1:

post.sort\_name()

elif key == 2:

post.sort\_like()

elif key == 3:

post.sort\_cost()

elif key == 4:

post.printlist()

check = int(input("Введите:\n1 - Если хотите сохранить в файл\n0 - Если не хотите сохранять в файл\n"))

post.save\_file(check)